

# Public Transportation Efficiency Analysis

## Phase 4 project

Designing dashboards and reports in IBM Cognos to visualize on-time performance, passenger feedback, and service efficiency metrics can be a multi-step process. Here's a high-level outline of how to approach this task:

### 1. Data Collection and Preparation:

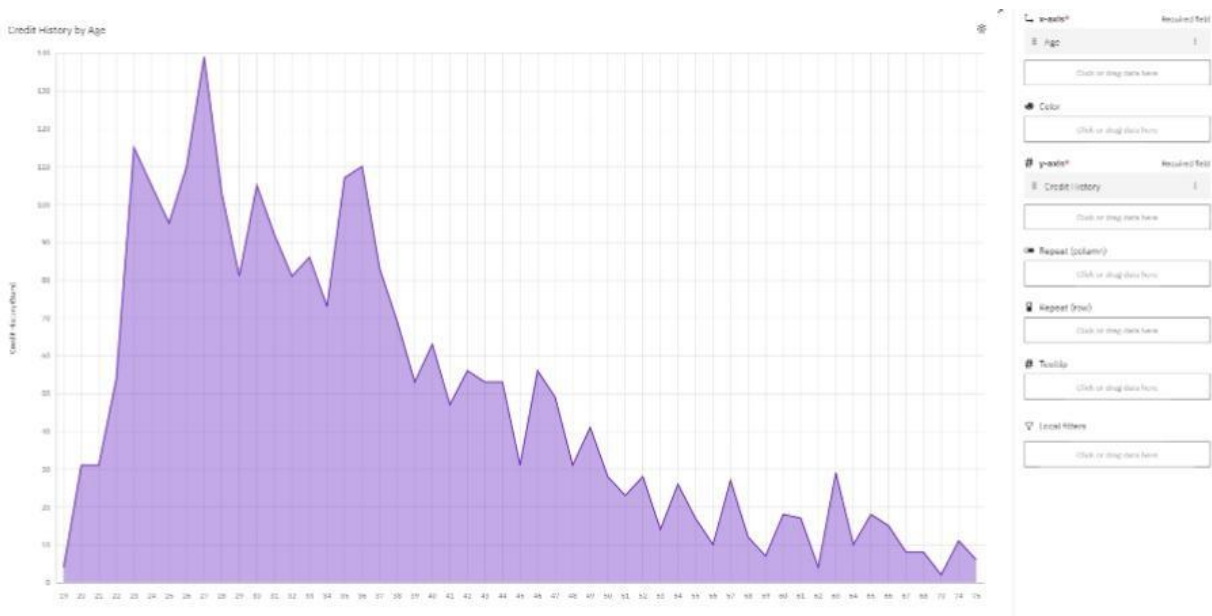
- Gather data on on-time performance, passenger feedback, and service efficiency metrics. Ensure that the data is in a structured format, such as a database or spreadsheet.

### 2. Data Integration in IBM Cognos:

- Connect to your data sources within IBM Cognos. This might involve setting up data connections, creating data modules, or importing data.

### 3. Dashboard Design:

- Create a new dashboard in IBM Cognos.
- Add visualizations like charts, graphs, and tables to represent on-time performance, passenger feedback, and service efficiency metrics.
- Customize the layout, colors, and styles to make the dashboard visually appealing and informative.



#### 4. Report Creation:

- Develop reports that provide detailed insights into each metric. These reports can be part of the dashboard or separate documents.

## 5. Data Analysis with Python:

- Use Python for advanced data analysis, which is not directly available in Cognos. For example, you can use libraries like Pandas, NumPy, and Natural Language Processing (NLP) libraries for sentiment analysis.

- Calculate service punctuality rates by analyzing the on-time performance data.

- Perform sentiment analysis on passenger feedback to gauge satisfaction.

Calculating Service Punctuality Rates:

Import pandas as pd

**# Assuming you have a DataFrame 'df' with columns 'arrivals' and 'departures'**

**# 'arrivals' and 'departures' represent arrival and departure times, e.g., 'HH:MM:SS'**

**# Convert 'arrivals' and 'departures' to datetime objects**

**Df['arrivals'] = pd.to\_datetime(df['arrivals'], format='%H:%M:%S')**

```

Df['departures'] =
pd.to_datetime(df['departures'],
format='%H:%M:%S')

# Calculate punctuality rates (assuming a certain
tolerance, e.g., 15 minutes)

Tolerance = pd.Timedelta(minutes=15)

Df['on_time'] = (df['departures'] - df['arrivals']
<= tolerance).astype(int)

Punctuality_rate = df['on_time'].mean() * 100

Print(f"Punctuality Rate: {punctuality_rate:.2f}%")

```

## 6. Integrating Python Code:

- To integrate Python code with IBM Cognos, you can use external data sources or create data modules that incorporate the results of your Python analysis.

### **Sentiment analysis on passenger feedback:**

#### **Import nltk**

```

From nltk.sentiment.vader import
SentimentIntensityAnalyzer

```

```

# Download NLTK's VADER lexicon

```

```
Nltk.download('vader_lexicon')
```

```
# Initialize the VADER sentiment analyzer
```

```
Analyzer = SentimentIntensityAnalyzer()
```

```
# Example feedback data
```

```
Feedback = "The service was excellent, but the food  
was terrible."
```

```
# Get sentiment scores
```

```
Sentiment_scores = analyzer.polarity_scores(Feedback)
```

```
# Interpret the sentiment scores
```

```
If sentiment_scores['compound'] >= 0.05:
```

```
    Sentiment = "Positive"
```

```
Elif sentiment_scores['compound'] <= -0.05:
```

```
    Sentiment = "Negative"
```

```
Else:
```

```
    Sentiment = "Neutral"
```

```
Print(f"Sentiment: {sentiment}")
```

```
Print(sentiment_scores)
```

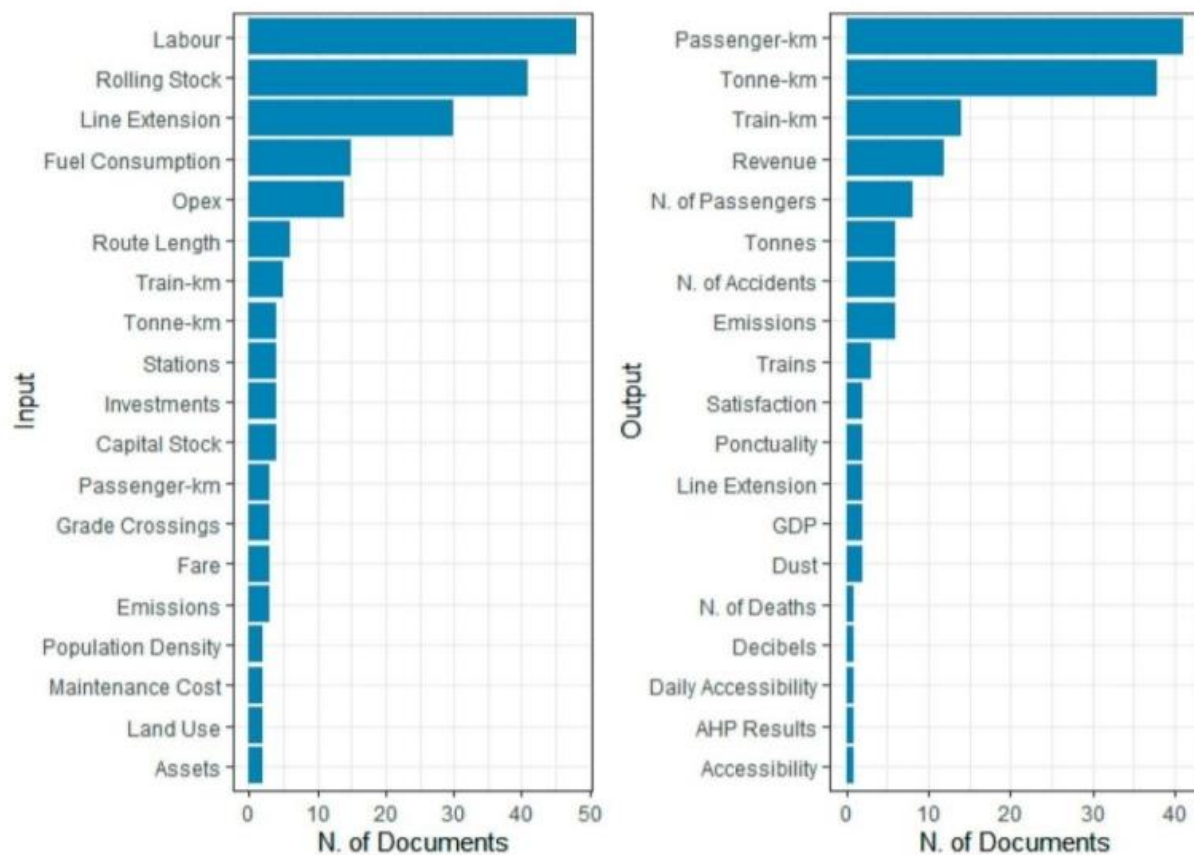
## 7. Automation and Scheduling:

- If you want to regularly update your reports and dashboards, consider automating the data retrieval and analysis processes using tools like cron jobs or IBM Cognos scheduling features.

## 8. Testing and Validation:

- Thoroughly test your dashboards, reports, and Python code to ensure that the data is accurate, and the visualizations are working as expected.

### **Dashboard:**



## 9. Documentation and User Training:

- Document the process, including how to refresh the data and run the Python code. Provide training for users who will interact with the reports and dashboards.

## 10. Feedback and Iteration:

- Gather feedback from users to make improvements and iterate on your dashboards and reports as needed.